Data-Centric Engineering (2026), 7: e26
doi:10.1017/dce.2026.10060

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An automated framework to characterize crack patterns in existing RC bridges

Angelo Cardellicchio[1], Vito Renò[1], Agnese Natali[2], Vincenzo Mario Di Mucci[3], Andrea Nettis[3], Sergio Ruggieri[3] and Giuseppina Uva[3]

[1]STIIMA CNR Bari, Italy
[2]Department of Civil and Industrial Engineering, University of Pisa, Italy
[3]DICATECH, Politecnico di Bari, Italy
Corresponding author: Vito Renò Email: vito.reno@cnr.it

Received: 12 January 2026 UTC; **Revised:** 23 April 2026 UTC; **Accepted:** 11 May 2026 UTC

Keywords: artificial intelligence; cracks; deep-learning; existing RC bridges; visual inspections

Abstract

The paper presents a framework to automatically identify crack patterns and the related features in existing reinforced concrete (RC) bridges. The challenge of this work is to define a tool for detecting the focused defect and highlighting the number and the orientation of cracks, allowing for correct interpretation and driving further evaluations on the residual life of the structure. The study is framed within the increasing interest in monitoring the structural health of existing bridges through automated tools, able to support engineers in the phase of visual assessment and interpretation of structural defects. When dealing with periodic inspection of large bridge portfolios, the support provided by automated tools can be fundamental for planning further strategies aimed at ensuring the structural safety and preventing future disasters. Given a stack of photos of a bridge structural element, an image stitching procedure is proposed to produce a near-complete image of the entire element. On the latter, a pipeline of deep-learning (DL) algorithms is employed to automatically detect and identify cracks (as a combination of object detection and segmentation algorithms). Finally, the proposed tool extracts cracks for counting and defines their orientation (i.e., vertical, horizontal, diagonal), in order to provide near-complete information about the crack pattern for the structural element. A full description of the methodology and the proposed algorithms is reported throughout the manuscript, showing the main pros and cons and assessing the effectiveness of the tool on a real-life case study.

Impact statement

The manuscript presents an artificial intelligence-based methodology for defining overall crack patterns in structural elements of existing reinforced concrete (RC) bridges, enabling the characterization of features like counting and orientation, after detection. The article provides the pipeline of a near-complete tool, named DOC-Crack (DOC stands for detecting, orienting, counting), characterized by different novel sub-modules to (i) postprocess a stack of photos to define a comprehensive view of the structural element under investigation; (ii) define, test, and apply combined object detection and segmentation algorithms to recognize cracks; (iii) derive key features of crack pattern, that is, orientation (horizontal, diagonal, and vertical) and counting. The application of the tool to a real-life case study shows that the obtained results, which can strongly support structural engineering domain experts in assessing the current state of existing bridges through visual inspections, allow for predicting possible future issues related to crack evolution and to plan related interventions with a

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

view to risk mitigation. The obtained outcome shows that the proposed tool is set to determine finer evaluation than existing approaches and to characterize a complete crack pattern, showing the precise extent of the defects over the entire structural elements.

1. Introduction

The maintenance of existing bridge portfolios is a primary issue addressed to public institutions, which need to ensure the safety of their infrastructures in order to prevent the repetition of past disasters (e.g., the collapse of the Polcevera Viaduct (Bazzucchi et al., 2018; Calvi et al., 2019) that have caused irreparable losses. Usually, bridges are subjected to multiple sources of hazards, such as seismic actions (Nettis et al., 2022), geological movements (Nettis et al., 2023; Calò et al., 2024), floods (Anisha et al., 2022), or natural decay of materials (Nettis et al., 2024). Therefore, the definition of the safety of existing bridges must be approached as a multirisk problem, following specific protocols, driving road management companies to employ reliable risk mitigation strategies. Taking as a reference the Italian case, the Ministry of Transportation, supported by the scientific community, developed and released new specific guidelines for the management and evaluation of the safety of existing bridges (MIT, 2021). These latter are characterized by a multilevel approach aimed at evaluating the health state of national bridge portfolios first, and then identifying the worst cases to employ specific interventions. The proposed approach involves six levels of assessment, each with different degrees of detail and complexity. The first three levels (Levels 0, 1, and 2) focus on a preliminary screening of bridge portfolios and are elaborated for defining prioritization lists, while the second three levels (Levels 3, 4, and 5) drive actions for bridges presenting higher risk in the earlier phases.

In this framework, one of the most significant phases is Level 1, during which onsite inspections of bridges are performed to assign an overall health score based on a detailed visual inspection of structural elements, including decks, girders, piers, bearings, and abutments. In this phase, the potential impact of various risk sources, including traffic, floods, landslides, and earthquakes, is also evaluated. The above tasks are addressed by trained surveyors, who conduct onsite inspections to record observed defects, including their intensity and extent, using a specific form. To complete the assessment, surveyors assign a score and take photographs, with the aim of tracking the evolution of each defect over time (i.e., by performing periodic inspections). On the one hand, the onsite visual inspection is a direct and practical method for detecting defects; on the other hand, several issues can impact the reliability of the final evaluation. In fact, one major issue is the subjective interpretation of defects by surveyors, which can be affected by human factors, such as attention lapses and fatigue, especially when dealing with a large number of bridges in a short time. In addition, external conditions, such as weather, lighting, and the distance between the surveyor and the inspected element, can also introduce uncertainties in defect scoring.

The matter becomes more complicated when assessing cracks, considering this type of defect is more difficult to survey and, at the same time, extremely important for defining the structural health of existing bridges. In this context, crack patterns are of interest and can be defined as the arrangement, shape, orientation, and distribution of cracks on the surface of the structural element. In other words, a crack can be visually considered the spatial aggregation of structural discontinuities detected on a concrete surface; from the perspective of an automated computer vision (CV) system, a crack is a continuous cluster of connected foreground pixels identified by a binary segmentation mask. The assessment of the evolution over time of crack patterns can be considered the most intuitive form of damage assessment and provides insights into the structural strength and durability of RC structures. Different aspects should be considered when discussing cracks and their related patterns. Given a generic RC structural element, the first aspect to address is the number of cracks, which reveal the extension of the defect within the element. Afterward, the extent of each crack should be assessed by evaluating its width and length to define the deterioration state of the structural element. Finally, another important aspect concerns the orientation of cracks (e.g., horizontal, vertical, and diagonal), which can reveal the causes of their formation and enable planning

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

further action to minimize potential risks. By juxtaposing the above information, a general understanding of the problem can be developed by defining an overall crack pattern, which can provide insights into ongoing phenomena.

On this basis, it is clear that there is a need to develop new tools that support surveyors during onsite inspections, and in particular to aid in the characterization of the most delicate defects, such as cracks (Spencer et al., 2019). This paper addresses this issue by proposing a tool, DOC-Crack, an artificial intelligence (AI)-based framework for detecting, orienting, and counting cracks (according to the proposed acronym) in existing RC bridges. In detail, the proposed approach involves taking a stack of photos of the structural element to be inspected (e.g., using a high-resolution camera or an unmanned aerial vehicle, UAV) and stitching them to create a near-complete panoramic image of the element. Afterward, a DL pipeline is proposed to accurately detect the defective regions and segment their traces. By analyzing such traces across the stitched panorama, the framework extracts the complete crack pattern, enabling the automated counting and orientation assessment of all cracks. The proposed method was developed as a supervised DL approach, for which the authors collected and labeled a dataset by hand. DOC-Crack was developed and tested on a real case study, showing the high potential of the tool in supporting the onsite operations of surveyors and driving future actions for disaster prevention.

Overall, the main novelties of the proposed approach are the following.

- First, the modular integration of a stitching algorithm allows for the investigation of cracks over the entire structural element rather than confining the analysis to limited parts of specific images. Consequently, this enables inspection of composite image collections representing large structural elements of RC bridges, resulting in fewer distortions and artifacts.
- The framework adopts a modular approach to object detection and segmentation. Therefore, both the detection and segmentation modules can be replaced if required. Subsequently, a novel segmentation approach was developed to precisely trace the cracks. Specifically, starting from the bounding box returned from the detector, the segmentation module employs a pixel-level binary classification to identify thin cracks. This explicitly preserves the highest spatial resolution, preventing the loss of ultra-thin, continuous crack topologies that often disappear during the aggressive max-pooling and downsampling stages of standard encoder–decoder networks such as U-Net.
- The computation of crack orientation assumes paramount importance, serving as an effective tool to support domain experts in assessing the current condition of the structural elements.

It is important to emphasize that the novelty lies mainly in the end-to-end integration and workflow application. In other words, the main contribution lies in developing a modular framework that provides a structural mapping from raw UAV imagery. By integrating CV and DL, the proposed pipeline translates pixel-level predictions into meaningful insights, providing users with a comprehensive view of the structural health of RC bridges and serving as an effective decision-support system.

The rest of the paper is organized as follows: Section 2 reports the background about cracks characterization through CV-based approaches; Section 3 proposes the complete description of DOC-Crack; Section 4 reports the numerical experiments for training the proposed methodology; Section 5 proposes the application of DOC-Crack for a real case study; Section 7 provides the conclusions and the further developments.

2. CV-based approaches for characterizing cracks in RC elements

The characterization of cracks in RC structures is a fairly discussed topic in scientific literature, which in the last decades has proposed and developed several methodologies for detecting cracks from images and extracting specific information for better characterizing structural conditions. Generally, several methodologies to detect cracks could be mentioned, such as ultrasonic waves, stereoscopic camera systems, or laser scanners (Ding et al., 2023). Nevertheless, of interest for the aim of this paper is to provide a background about CV approaches for the identification of cracks in RC structures directly from images,

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

objects, and pixels. In this field, four main approaches can be mentioned: classification, segmentation, feature detection, and object detection (Di Mucci et al., 2024; Guo et al., 2024; Laflamme et al., 2026).

Classification consists of categorizing objects or patterns in images according to some predefined classes. The aim is to define and assign a specific class to an input image or to a part of it. For example, Liang (2019) proposed a three-level image-based approach for performing classification and localization of damages in RC bridges, focusing on data derived from postdisaster inspections. Kim et al. (2022) proposed a classification method to map crack pixels on images taken from smartphones and accounting for different angles, with the aim of characterizing the localization of cracks.

More suitable for detecting crack features in RC bridges is the segmentation, which is an approach in which the image is subdivided into multiple segments, with the aim of simplifying the identification of the regions to define in the image. Generally, segmentation allows the location of objects and the definition of boundaries in the focused image. In this regard, Li et al. (2019) proposed a segmentation approach for proposing a multidefect detection (among which are cracks) based on pixel-level through the development of a fully convolutional network. A database of 2.750 images was collected, and the proposed approach achieved an accuracy of 98%. Saleem et al. (2020) proposed an approach exploiting instance segmentation through DL to localize and classify cracks in bridges, by using inspection images from UAVs. Ding et al. (2023) developed an approach using UAV surveys to detect and quantify cracks in RC without reference markers and using an independent boundary refinement transformer. Deng et al. (2023) proposed a three-dimensional (3D) crack mapping to localize cracks in a surface by using a point cloud model developed on a video-based reconstruction. Chu et al. (2022) proposed Tiny-Crack-Net, a multiscale feature fusion network with attention mechanisms to solve the problem of class imbalance in crack segmentation. The proposed approach achieved an accuracy of 91% for cracks with a width of 0.05 mm. Other works using segmentation can be mentioned for the sake of brevity, such as Wang and Su (2022), Kang and Cha (2021), Wang et al. (2020).

Another approach for crack definition is feature detection, which allows identifying and locating patterns (i.e., features) in the focused images. The approach is suggested for defects like cracks, since feature detection can identify both simple lines and complex figures. If applied in different moments, feature detection can also allow for investigating the variations of defects over time. For example, Liu et al. (2019) proposed an image-based crack method aimed at extracting features from bridge piers by using a UAV and 3D scene reconstruction. From the proposed method, the authors were able to define the shape of the cracks and the related width, with negligible errors.

The last and more recent CV-based method for cracks (and defects) characterization is object detection, able to recognize objects in images. In fact, it consists of identifying, localizing, and classifying the desired object within the image. Two main kinds of object detectors are widely spread in practice: (a) two-stage object detectors, which present two neural networks able to identify and classify, respectively, objects in the image; (b) single-stage detectors, which adopt a single network to address both localization and classification tasks. Regarding cracks in RC elements, Yu et al. (2021) proposed a real-time detection of cracks in bridges by combining the potentialities of UAVs and a modified version of YOLOv4. This latter was modified by using focal loss to optimize the loss function. The above modification improved accuracy, achieving 97.6%. Cha et al. (2017) proposed a convolutional neural network (CNN) based on a sliding window technique for detecting cracks, showing an overall accuracy of 98% by training the network on 40.000 images. Deng et al. (2020) proposed a version of a Faster R-CNN on a dataset constituted by 5.000 images to detect cracks in RC bridges, accounting for interferences due to handwriting automatically. The obtained results were compared with those obtained by YOLOv2, showing several advantages in terms of accuracy. Park et al. (2020) used YOLOv3 in a real-time defect detection of cracks in RC structures, for characterizing the size and position of the considered defect. The results of experiments demonstrated high accuracy (about 94%) and precision (about 98%). Other works can be mentioned about the use of object detection for cracks, such as Yu et al. (2021), Jiang and Zhang (2019), but it is worth noting that most of the above works, although present approaches with high accuracy, are mainly focused on the characterization of single cracks (also very thin) and usually exploit single images given by onsite inspections.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Instead, the proposed approach, DOC-Crack, potentially leads to a major advance in the field of crack detection, providing an end-to-end framework to reconstruct the complete crack pattern in RC structural elements, starting from unlabeled imagery directly gathered in the field. In other words, DOC-Crack extracts a complete semantic assessment of the status of the RC structural element, providing the domain expert with human-readable information to localize, count, and estimate the orientation of each crack. To achieve this goal, DOC-Crack was engineered as an end-to-end framework that exploits a modular pipeline to provide the desired information from the imagery of the RC structural element under investigation.

3. DOC-Crack: Definition of the methodology and description of sub-modules

This section presents DOC-Crack, describing the proposed submodules with the related datasets and algorithms. The framework of DOC-Crack is shown in Figure 1, which reports the complete processing pipeline at the base of the method.

The main steps are briefly summarized, and are accurately described in the next Sections:

1. **Onsite inspection and dataset collection**: to perform onsite surveys by domain experts on structural elements. Ideally, the surveyor should take photos in a specific way, only possible by exploiting proper equipment, such as UAVs. Still, to collect additional data to train the automatic algorithms of detection, an existing database or other photos (i.e., closer to cracks) should be collected.
2. **Stitching**: the collected photos are elaborated through a stitching procedure, providing a comprehensive view of the structural element under investigation.

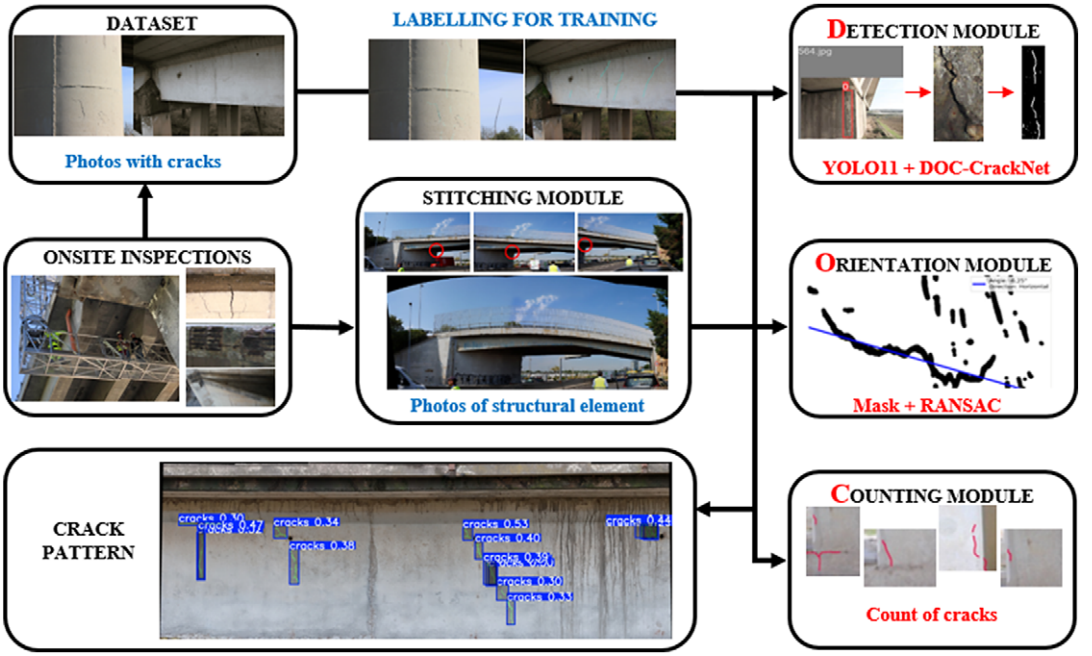


Figure 1. *The DOC-Crack framework. Once a dataset is gathered and prepared for training, a detection network based on stable iterations of YOLO is trained. Then, during inference, partially overlapped images of RC bridges are stitched together, and a method containing an object detection pipeline, followed by an instance segmentation tool, is used to provide a preliminary assessment of the cracks within the RC bridge.*

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

3. **Labeling**: the collected dataset must be labeled for the purpose of CV algorithms. In particular, the labeling should identify the exact trace of cracks.
4. **Crack detection, counting, and segmentation**: parts of the image containing cracks are detected and counted by using specific object detectors based on YOLO (Redmon et al., 2016), and trained on the above dataset to improve the efficiency of the algorithm. Once the crack is identified, a novel segmentation technique is employed to accurately define the crack patch.
5. **Crack pattern and orientation**: the identified cracks can be relocated on the initial stitched image to define the crack pattern. In addition, the orientation feature can be estimated by exploiting a pixel-level evaluation procedure.

3.1. Onsite inspections and data collection

The first phase of the proposed procedure consists of performing onsite inspections, aimed at collecting data. Two types of data need to be gathered: (a) photos of the structural elements to be analyzed in DOC-Crack; (b) photos of cracks from different views to enrich the overall dataset for the training.

Concerning the first type of data, once the structural element to investigate is identified, two simple approaches could be exploited: (a) use of a UAV; (b) a human survey with a high-resolution camera. When using UAVs, a specific protocol should be followed. In particular, once the absence of obstacles has been assessed, the UAV performs the flight following a series of reference points. The latter are designed to create a virtual fence area, which the UAV follows according to a linear route, parallel to the longitudinal development of the structural element. The distance from the UAV to the structural element is estimated according to the optical cone of the machine and its resolution, by ensuring an overlap between consecutive photos of at least 30% and ensuring the same angle (as suggested in Wang et al. (2023)). Similar performance could be achieved through a high-resolution camera managed by human surveyors, as indicated above. Nevertheless, although this second option may be cheaper than the first, it is strongly influenced by the presence of obstacles (e.g., vegetation), the photographer's skill, and the height of the structural element being investigated (i.e., the deck and its components). A possible solution to the above issues is represented by the by-bridge platform.

Regarding the second type of data, a consistent dataset of defects on different structural elements must be collected, for example, (Cardellicchio et al., 2023; Ruggieri et al., 2023, 2025), ensuring that the minimum requirements for the subsequent steps are met. In particular, the collected dataset should be characterized by (a) scale, (b) diversity, and (c) quality. Regarding scale, as reported by Cardellicchio et al. (2022), a consistent size of dataset, characterized by several examples and typology of defects, should be collected. In addition, data balance should be ensured. Concerning diversity, the dataset should be characterized by enough intra- and inter-class variations, that is, each feature should be described by data accounting for variation of appearance, position, viewpoint, and pose, without discarding images with noise and occlusions. Finally, high-quality images are required, since low-resolution data can negatively influence the accuracy of the detection models.

3.2. Robust image stitching for global bridge view

Limitations of the sensing equipment, mainly in terms of field of view and resolution, imply that a single, comprehensive view containing the structural element to monitor is often not gathered in a single image. Therefore, providing a robust and reliable stitching procedure is a mandatory prerequisite for analyzing large-scale defects and assessing the overall structural health status. To this end, this work incorporates a well-assessed, feature-based stitching methodology, whose results are shown in Figure 2.

The pipeline begins by detecting and describing keypoints by means of a Harris corner detector (Harris et al., 1988), and local invariant descriptors, such as Scale-Invariant Feature Transform (SIFT) (Lowe, 2004), which were computed to match features reliably between partially overlapping images. Once possible matching points are computed, the Random Sample Consensus (RANSAC) algorithm (Fischler and Bolles, 1981) was used to estimate a homography matrix. The primary motivation for the selection of

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

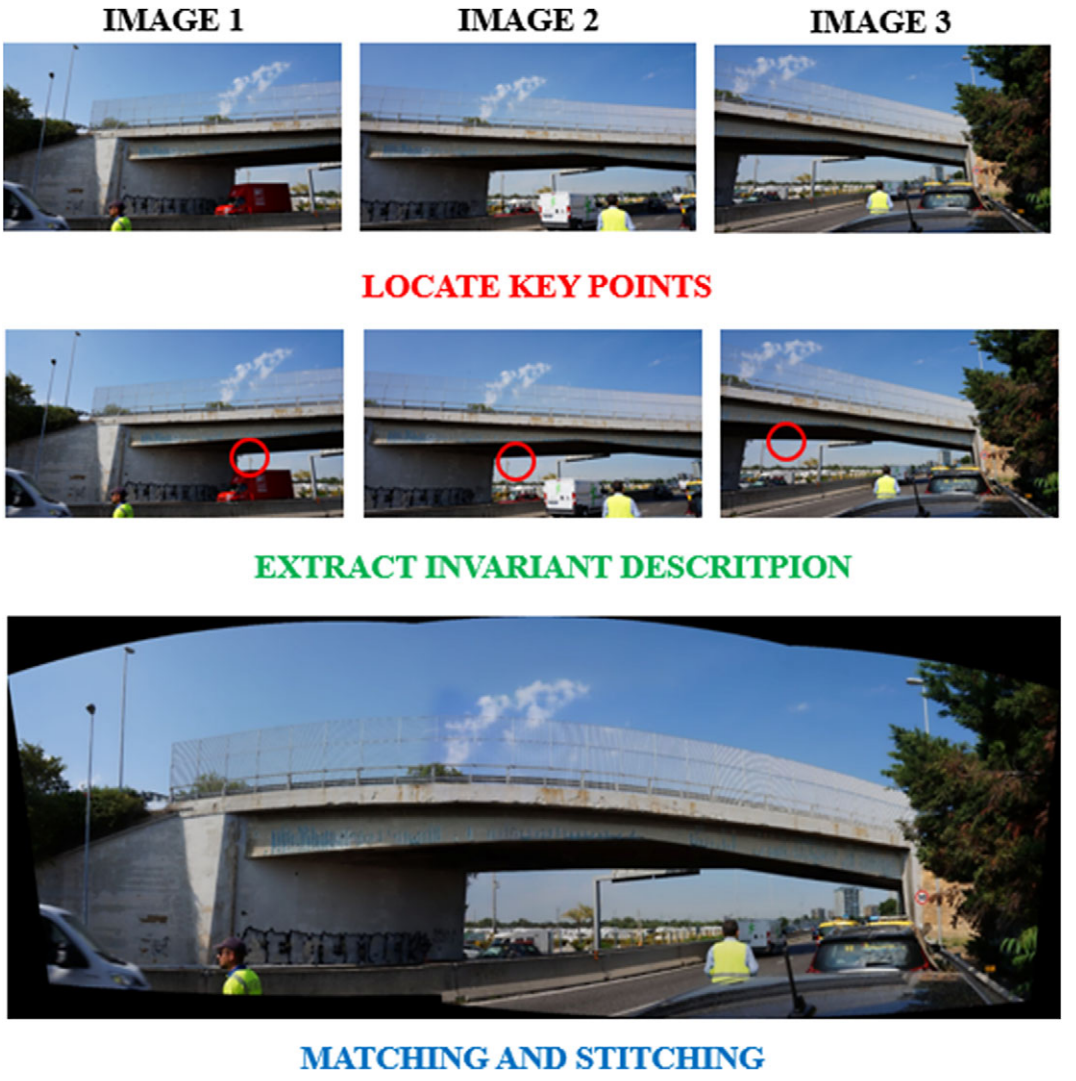


Figure 2. *An example of image stitching. First, key points are located using the Harris detector. Then, invariant descriptors are extracted and matched for each key point to identify overlaps between images. Finally, the panorama is obtained using a warping transformation.*

RANSAC is its robustness to outliers, which ensures that the estimated homography is based on geometric consensus, thus minimizing misalignments. Finally, the starting images were warped according to the homography matrix and blended to create a seamless overall view of the structural element.

3.3. Dataset and preparation for detection algorithms training

The data collected following the procedure highlighted in Section 3.1 were used to create two different datasets. The first dataset, the *large-scale defect* dataset, was used to train a deep neural network for the crack detection task, that is, localizing areas of interest within the images where a crack is likely to be found. The second dataset, the *high-precision crack segmentation* dataset, was then used to train another neural network to perform semantic segmentation, thereby identifying cracks within the areas of interest identified by the detection algorithm.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

3.3.1. Dataset for large-scale defect detection

This dataset was used for training a DL object detector to identify several classes of damage. The dataset was composed of 10.779 images, which were manually labeled by domain experts using the *Computer Vision Annotation Tool* (CVAT) (Sekachev et al., 2020). Specifically, the labeling was performed to deal with the damage detection task; therefore, the outcome was in the form of bounding boxes. The labeling allowed the identification of seven types of different defects, which were used to provide multiple damage detection (Ruggieri et al., 2024; Ruggieri et al., 2025).

3.3.2. Dataset for high-precision crack segmentation

The dataset for training the segmentation network originated from a set of 450 images, specifically selected for their high-quality representation of crack defects. This dataset was labeled by domain experts by hand-tracing the full extent of visible cracks. Also in this case, the CVAT tool was used, and the results were exported in the Dataset Management Framework (Datumaro) format (Datumaro, 2022).

After labeling, a patch-based generation method was exploited to create the training samples for the pixel-wise classifier. Specifically, a square filter of size F was used to extract a patch at each pixel location; each of these patches was then labeled as either a *crack* or a *noncrack* sample, based on the fact that its central pixel belonged to a crack or not. These generated $M \times N \times 450$ data samples where $M \times N$ was the size of each image. Thus, this approach addresses data scarcity by generating data directly from the source pixel rather than through image-processing transformations, thereby mitigating the risk of overfitting. However, this process implied a severe class imbalance, with the *noncrack* images vastly outnumbering the *crack* samples. As such, a hard-negative mining sampling strategy was implemented to balance the data and improve performance.

Still, before generating the patches, the 450 images were randomly split into two distinct sets, namely a training and a validation set, using a 70−30 split. This was specifically designed to prevent data leakage and thereby improve the overall generalization capabilities of models trained on the dataset. Overall, this resulted in three datasets of roughly the same size, as detailed in Table 1.

3.4. Crack detection and counting

The enabling step in the processing pipeline was provided by the object detection model, which was specifically trained to identify multiple defects within imagery representing structural elements of RC bridges. To this end, starting from the work in Ruggieri et al. (2025), two variants on the basic architecture of YOLO11 Jocher and Qiu (2024) were proposed. Specifically, the YOLO family was originally proposed by Redmon et al. (2016) as a single-stage object detector, that is, a DL model capable of localizing and classifying objects of interest within an image using a single CNN. In Ruggieri et al. (2025), the concept of Convolutional Block Attention Module (CBAM) was applied to the neck of YOLO11, highlighting improved detection capabilities for the network. While improving the overall performance of the network, the use of CBAM poses some constraints; specifically:

- **Feature Interaction Limitations**: CBAM decouples attention into the *channel* (i.e., monodimensional) and *spatial* (i.e., bidimensional) components. While effective, this approach prevents the

Table 1. *Datasets created for high-precision crack segmentation*

Image patch size	# of crack patches	# of noncrack patches
64 × 64	195.971	198.656
128 × 128	195.813	198.656
256 × 256	194.376	198.656

Note. The noncrack patches are negatively undersampled to provide a balanced dataset. The number of crack patches varies with the size of the considered patch; as expected, larger patches yield slightly fewer data samples. No padding was applied as border patches were not considered.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

capture of true 3D attention weights, which are given by relationships between the spatial dimensions and the channels. Consequently, distinctive features restricted to specific spatial locations or feature channels may be suppressed due to the global pooling performed by the CBAM attention, potentially leading to information loss.

- **Reliance on Heuristics and Parameters**: The CBAM module was designed around a set of heuristics, such as specific kernel sizes and reduction ratios in its multilayer perceptron (MLP), thus introducing additional hyperparameters and computational overhead, increasing the complexity of model tuning and inference.

To address these issues, two attention mechanisms were separately explored. First, the *Global Attention Mechanism* (GAM) introduced in Liu et al. (2021) was considered to address the limitation of feature refinement by focusing on retaining information across dimensions to strengthen interactions. The main innovation behind the GAM lies in its design, which minimizes information reduction while maximizing cross-dimensional feature interaction to enhance global interactive representation. GAM achieves this by utilizing sequential submodules, notably in its Channel Attention submodule, where the input tensor is permuted and then processed by an encoder–decoder MLP to learn channel-spatial dependencies.

The second explored mechanism was SimAM, introduced by Yang et al. (2021), which leverages a fundamentally different approach inspired by neuroscience principles, allowing for the calculation of full 3D attention weights in an efficient, parameter-free manner. Specifically, the SimAM mechanism is derived from the neuroscientific theory of spatial suppression, which states that the most informative neurons are those that display distinctive firing patterns within a neighborhood and, therefore, should be prioritized. To this end, SimAM computes the importance of a single target neuron n by defining an energy function e_n that measures the linear separability between n and its neighbors. A lower energy value indicates that the neuron is more distinctive and, therefore, more important for visual processing; consequently, the attention weight is computed as the inverse of the energy $\sigma(1/e_n)$.

Two specific sections of the YOLO11 original network were targeted by the insertion of these mechanisms, namely the backbone and the neck. As for the backbone, two reasons underlie this choice: the first is to provide a mechanism that forces the network to recalibrate feature weights early on during training, effectively dimming the background and non-relevant information while highlighting the foreground elements of the image. Another effect is a prioritization of the channels that respond to specific spatial frequencies, such as the edge of a crack, over generic texture channels. Regarding the neck, it is worth noting that this section of the network combines semantic features extracted at various abstraction levels, enabling multiscale detection. Inserting attention mechanisms in this part of the network helps in achieving two goals: first, when features from a deep layer are concatenated with those from a shallow layer, a semantic gap can arise; attention mechanisms can help in deciding which details from the shallow layer are actually relevant to the semantic representation provided in the deep layer. Second, since deep layers inherently lose spatial precision, inserting attention in the neck helps recover precise defect localization from the shallow layers.

The models were first trained from scratch on the detection dataset, split according to a 70−30 strategy. Consequently, the trained models could be used to localize cracks within the input imagery, allowing for the estimation of defect positions within the final stitched image. In addition, the use of the object detector enables the counting of the number of occurrences of the defect. To this end, a counting module was implemented to quantify all cracks across the stitched image, thereby evaluating the extent of the crack pattern.

3.5. Crack segmentation

Once the cracks were detected using the best-performing object detection model, a novel pixel-based approach was developed to estimate the pattern of the identified cracks. This approach was based on a specifically designed neural network, named *DOC-CrackNet*, aimed at framing the problem in a different and possibly simpler way. Specifically, instead of using classic segmentation approaches based on

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

U-shaped networks, the idea was to frame the problem as a *binary classification* problem, thereby training a simple CNN to distinguish between areas whose center *likely belongs to a crack* and areas whose center does not. This allowed three main improvements over other existing approaches:

1. The problem of data availability was solved, as it was possible to extract a large number of patches from a starting image. Ideally, given an image of $M \times N$ pixels, the proposed approach could extract up to $(M - fracW2) \times (N - fracW2)$ different patches, where W represents the size of the selected patch. This allowed an exponentially higher number of samples to populate the dataset for the investigation.
2. The problem of interpretability was partially mitigated by the adoption of simpler models, such as the one used in this work, which was composed of only three stacked simple convolutional layers.
3. Framing the problem as a binary classification allowed a probabilistic output that, in future works, can be exploited to further improve the overall results of the network by adopting a flexible, variable classification threshold.

Interestingly, it must also be noted that exploiting simpler models allows for the reduction of the overall computational weight of the network, potentially enabling the adoption of this approach even on constrained hardware in real-time. The feature map size-halving criteria established by (He et al., 2016) was followed to design DOC-CrackNet. The network scheme of DOC-CrackNet is shown in Figure 3.

The latest layer of DOC-CrackNet presents a sigmoid activation function, which was chosen over the softmax activation function as the problem is framed as a binary classification task. In other words, through the sigmoid activation function, DOC-CrackNet provides a probability value $\sigma \in [0, 1]$ representing the likelihood that the patch is a positive (i.e., it represents a crack) or a negative sample (i.e., it does not represent a crack). This evaluation requires fixing a threshold value ϕ to distinguish between the two different cases and to provide the metrics used for the overall performance evaluation:

- **True Positives** (**TP**): the number of patches whose central pixel represents a crack and is correctly classified as a positive sample.
- **True Negatives** (**TN**): the number of patches whose central pixel does not represent a crack and is correctly classified as negative samples.
- **False Positives** (**FP**): the number of patches whose central pixel does not represent a crack and is incorrectly classified as a positive sample.

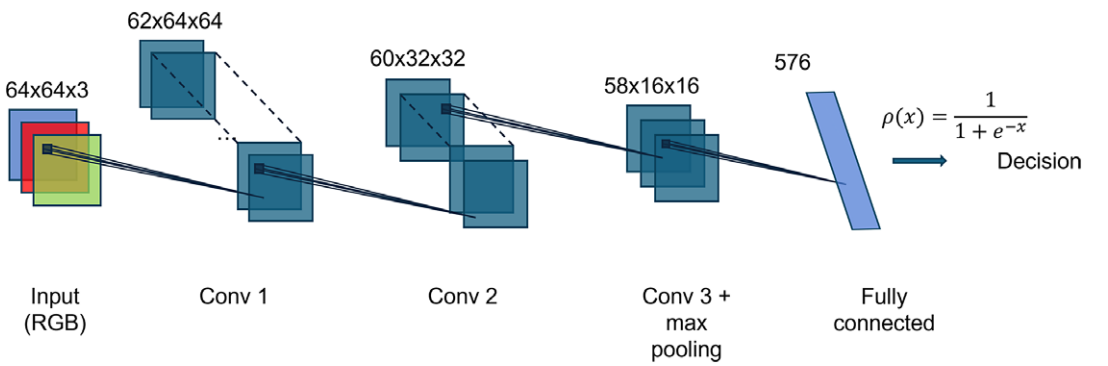


Figure 3. *The architecture of the proposed network. Specifically, DOC-CrackNet accepts an RGB image of the desired size (in this case* 64×64*). The image is then fed to three consecutive convolution layers, after which it is reduced in size using a max pooling layer. Finally, a fully connected layer extracts a feature embedding vector of 576 elements, fed to a sigmoid classifier for the final decision.*

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

- **False Negatives** (**FN**): the number of patches whose central pixel represents a crack and is incorrectly classified as a negative sample.

The threshold is manually set using threshold-moving (Provost, 2000) to maximize the value achieved by the F1 score, defined as:

$$F1 = 2 \cdot \frac{P \cdot R}{P + r} \tag{1}$$

where P represents the *precision*, defined as:

$$P = \frac{TP}{TP + FP} \tag{2}$$

while R is the *recall*, defined as:

$$R = \frac{TP}{TP + FN} \tag{3}$$

Therefore, higher precision is achieved when the model can misclassify a few negative samples, while higher recall is achieved when the model can misclassify a few positive samples. The F1 score synthesizes these aspects within a single metric.

The outcome of this step is a binary *estimation mask* M, whose value is *white* for the points where the crack is identified and black otherwise. This allows for identifying and segmenting each crack within the original images, delineating its pattern within the structural element.

Finally, it is worth underlining that the structure of DOC-CrackNet was designed to be flexible enough to accept patches of different sizes as inputs, allowing the evaluation of the impact of contexts for different sizes on the overall performance of the network.

3.6. Estimation of main crack orientation

Assessing the principal orientation of a crack is a critical feature for structural health management. In other words, while cracks often exhibit complex, nonlinear geometries, providing a robust, first-order approximation of their dominant linear trend (e.g., horizontal, vertical, or diagonal) can provide important insights into the causes behind a certain defect, as well as future maintenance strategies that can be proven effective.

The orientation is estimated in a two-stage process. First, the binary segmentation mask M obtained after crack segmentation is preprocessed by applying a morphological erosion filter. This step filters out segmentation artifacts and spurious branches, therefore ensuring the analysis is performed on the main component of the crack.

Once the erosion filter was applied M, the resulting erosion mask M_{er} was used as an input to train a regressor by means of the RANSAC algorithm (Fischler and Bolles, 1981). The motivation behind the use of RANSAC lies in the consideration that a simple linear regressor would estimate an overall trend that is extremely sensitive to outliers. Using a robust regressor would filter out residual segmentation noise, as well as the nonlinear portions of the crack, allowing for robust identification of the single best-fit line representing the dominant trend of the crack.

Once the optimal regression model R_i is computed using RANSAC, its coefficient is converted to an angle α and used to evaluate the direction of the crack according to the thresholds shown in Table 2. As can be observed, horizontal cracks are classified as ranging from 0° to 25°, diagonal cracks are classified as ranging from 25° to 60°, and vertical cracks are classified as ranging from 60° to 90°.

It is worth specifying that the proposed approach for estimating crack orientation is mainly based on the classification provided by the Italian guidelines MIT (2021), which subdivides cracks into the three classes defined above. Obviously, more complex crack shapes could be experienced in some real-life cases, for which the classes subdivision like the one provided could be too simplistic. Nevertheless, during

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Table 2. *Threshold for determining the orientation of the crack according to the slope of the estimated model R_i*

Orientation	Minimum value (degrees)	Maximum value (degrees)
Horizontal	0	25
Diagonal	25	60
Vertical	60	90

Note. For the sake of simplicity, only values between 0 and 90 degrees were considered. However, if an offset of plus or minus $\frac{\pi}{2}$ is applied to the specified values, the thresholds can be adapted to all the possible scenarios.

onsite inspections, the domain expert must follow the guidelines and rules and then needs to assume a class within the three classes above defined. Still, although DOC-Cracks allows to detect and to trace also complex cracks, the logic behind the tool (not only with reference to orientation, but in the near-complete definition of crack pattern) is to support domain experts, who should always validate the output provided by the proposed automatic tool. Finally, it is clear that more complex approaches could be employed to define orientation (e.g., instance-level skeletonization to map varying orientations along the continuous path of a single curved crack), which are beyond the scope of this paper and can be the object of future developments.

4. Experimental section

In this section, the experiments performed to elaborate the steps of the framework are described. All the experiments were carried out by using a machine equipped with an Intel Core i9-14900HK, 64 GB of RAM, and an NVIDIA GeForce 4090 with 24 GB of VRAM. The experiments were performed using Python as the primary programming language and the PyTorch library (Paszke et al., 2019) under a single-GPU configuration.

4.1. Training of object detector and automated crack segmentation

4.1.1. Full-scale RC bridge object detector

For the case at hand, the base dataset was gathered by following the procedure described in Cardellicchio et al. (2023); Ruggieri et al. (2023, 2025). Specifically, a total of 10,779 images were acquired during surveys of the structural elements of existing RC bridges, mainly decks, piers, pier caps, and abutments. Most of the gathered images were related to one or more structural elements, showing several defects of different extents and intensities. According to the above-mentioned works, the first labeling was performed to identify seven types of defects related to RC bridges.

The results of testing different configurations of YOLO models with various attention mechanisms are detailed in Table 3. Crucial considerations can be made concerning how placing different attention mechanisms in different parts of the network influences its behavior.

Specifically, the integration of GAM within the backbone yielded an overall superior value for mAP 0.5:0.95 outperforming baseline architectures by approximately 2.4% when the largest model scale (YOLO11x) was considered. This implies that, by filtering background noise at the earliest stages of feature extraction, the model achieves higher robustness, resulting in a stronger, noise-free feature representation. This enhanced feature space enables the detection head to focus solely on structural defects, resulting in tighter, more effective, and precise bounding boxes. Another significant finding is that the insertion of SimAM within the neck resulted in the optimal F1 score for the medium-sized model, thus implying a better trade-off between precision and recall, with a top value of 69.46% . This therefore suggests that the 3D attention weights proposed by SimAM are effective during the multiscale fusion stage, allowing for reliable discrimination of true defects from complex textures.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Table 3. *Comparison of model performance across metrics*

Model	Precision (%)	Recall (%)	F1 Score (%)	mAP 0.5 (%)	mAP 0.5:0.95 (%)
Baselines					
YOLOv11n	57.79	39.45	46.89	42.80	23.72
YOLOv11s	62.35	40.73	49.27	45.99	27.43
YOLOv11m	71.86	47.35	57.08	52.58	33.38
YOLOv11l	65.49	49.62	56.46	51.95	32.90
YOLOv11x	79.93	50.62	61.98	56.97	39.81
CBAM Ruggieri et al. (2025)					
YOLO11n	37.79	31.88	34.85	27.45	12.51
YOLO11s	52.87	45.05	48.65	45.44	25.57
YOLO11m	75.50	47.88	58.60	54.75	34.48
YOLO11l	77.02	50.82	61.24	56.37	35.46
YOLO11x	82.87	52.61	64.36	59.38	41.86
GAM (Backbone)					
YOLO11n	41.85	38.51	40.11	37.45	19.66
YOLO11s	80.25	50.58	62.05	56.00	36.33
YOLO11m	**90.76**	55.28	68.71	61.46	46.75
YOLO11l	86.77	**56.89**	68.72	**62.02**	**48.64**
YOLO11x	**91.35**	**55.97**	**69.41**	**62.43**	**51.83**
GAM (Neck)					
YOLO11n	46.63	36.26	40.80	35.99	18.52
YOLO11s	72.94	49.93	59.28	54.77	35.48
YOLO11m	84.36	55.37	66.85	61.03	45.47
YOLO11l	81.29	55.38	65.88	60.00	45.12
YOLO11x	88.32	55.35	68.05	60.93	47.76
SimAM (Backbone)					
YOLO11n	49.47	36.83	42.22	38.54	19.91
YOLO11s	80.79	52.61	63.72	56.76	37.61
YOLO11m	89.34	54.84	67.96	60.75	47.14
YOLO11l	83.89	55.40	66.73	60.51	46.57
YOLO11x	88.74	55.65	68.40	61.56	47.94
SimAM (Neck)					
YOLO11n	68.00	44.65	53.90	47.97	27.59
YOLO11s	87.24	55.05	67.50	60.33	41.78
YOLO11m	**90.87**	**56.21**	69.46	**62.16**	46.99
YOLO11l	89.38	55.40	68.40	61.59	47.68
YOLO11x	90.15	55.56	**68.75**	61.66	**48.65**

Note: Values in green indicate the best results, values in red the second-best, and values in orange the third-best.

It is worth framing the impact of the proposed architectural enhancements on different network sizes. Specifically, there is a larger contribution for smaller scales (i.e., nano and small), mainly with the parameters-free SimAM attention mechanism. This effectively compensated for the relatively limited capacity of such networks, showing relative improvements of nearly 15% when placed within the neck for YOLO11n. At the same time, the relative contribution of the attention mechanisms is less relevant (yet not negligible) for larger scales, mainly due to the high representation capability already embedded in those architectures. Consequently, a sweet spot can be determined for the medium-sized model, as demonstrated by the introduction of SimAM in the neck.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Overall, these findings suggest that using a combination of SimAM and GAM can provide further improvements and should be explored in future work.

4.1.2. Crack detection

The trained model was fine-tuned to focus on cracks on the second, smaller dataset. This was characterized by 450 images containing visible samples of cracks. For each image of this new subsample, a further annotation was performed by hand-tracing cracks. Some examples of the new labeling, along with their corresponding ground truth, are shown in Figure 4.

According to the procedure defined in Section 3.4, crack patches were derived, and the resulting segmentation dataset was composed of 195.971 positive samples and 198.656 negative samples. Some of the results are shown in Figure 5, in which patches with and without cracks are reported.

Subsequently, a data augmentation procedure was also considered, through a series of augmentation steps specifically tailored to retain the semantic meaning of each image. Specifically, random rotations



Figure 4. *On the right, the results of the labeling segmentation procedure on the image on the left are highlighted in light blue or pink.*

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

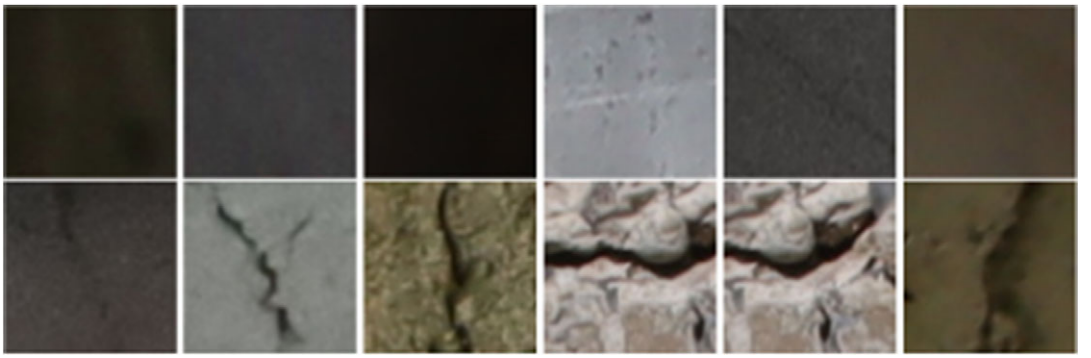


Figure 5. *A sample of the segmentation dataset. On the first row, six patches where cracks were not found are shown, while on the second row, six patches where cracks were found are reported.*

around the central pixel, random flipping (both horizontal and vertical), and random adjustment in terms of sharpening, contrast, and saturation were considered. All the augmentation steps were designed to provide a larger variability of the data fed to the network while retaining the information provided by the central pixels of the patch, and then not varying its class.

As for the hyperparameters, binary cross-entropy was selected as the loss function, with SGD as the optimization algorithm and 25 epochs of training with a learning rate of 0.01. All these parameters were experimentally found to be optimal for the problem at hand. Finally, as for the size of the input patch, let us recall that, as already stated in Section 3.4, DOC-CrackNet was designed to be flexible enough to accept patches of different sizes as input. To test the influence of the contextual information at different scales, the tests were conducted using square masks of three different input sizes, that is, 65×65, 129×129, and 257×257. These values were selected experimentally, considering the overall size of the images under investigation and the memory constraints of the hardware used during training. In the following, each version of DOC-CrackNet will be referred to using a suffix indicating the size of the input patch.

4.2. Optimal threshold and quantitative evaluation

To rigorously evaluate the framework and ensure there is no data leakage, the dataset was strictly partitioned at the image level prior to patch extraction. The performance of the segmentation module was quantified using Accuracy and F1 score. It is worth noting that standard full-mask spatial overlap metrics, such as the intersection over union (IoU) or the Dice coefficient, were not explicitly selected as evaluation criteria. This is due to the ultrathin morphology of structural cracks, which often span only a few pixels, and to the specific formulations of IoU and Dice coefficient in semantic segmentation, which make them highly sensitive even to single-pixel spatial shifts. In other words, a (structurally) valid crack prediction that is misaligned by a single pixel incurs an artificially severe penalty, rendering these metrics overly pessimistic for structural health monitoring. Therefore, the patch-based F1-score on the strictly separated dataset provides a more reliable representation of the framework's detection capabilities.

Consequently, an extensive ablation study was conducted to determine the optimal patch size, select from three sizes (64×64,128×128,256×256), evaluate the impact of data augmentation, and determine the optimal classification threshold σ, which was selected in the range [0.1,0.9] with a discrete step of 0.1. The results in terms of Accuracy and F1 score are reported in Tables 4 and 5, respectively.

According to the results in Tables 4 and 5, the architecture that significantly outperformed the others was DOC-CrackNet trained on 64×64 patches with data augmentation. If an optimal $\sigma = 0.5$ was selected, this model achieved an accuracy of 90.18% and an F1-score of 90.28% . This performance discrepancy is directly related to the receptive field provided by each patch dimension. Because DOC-CrackNet classifies the central pixel based on the surrounding window, the 64×64 patch provides sufficient local morphological context to recognize the linear continuity of a crack without overwhelming the network

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Table 4. *Accuracy of DOC-CrackNet for all the tested patch sizes and with or without augmentation*

Size	Augm	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
64	**No**	78.97	83.13	85.18	86.33	87.26	87.90	88.27	88.43	88.01
64	**Yes**	80.44	86.10	88.76	89.95	90.18	89.60	87.96	83.79	75.56
128	**No**	64.57	70.30	73.20	75.13	76.55	77.43	78.19	78.43	77.70
128	**Yes**	61.91	69.09	73.96	77.07	78.89	78.18	71.20	61.34	52.37
256	**No**	75.21	80.00	81.90	83.01	83.59	83.90	83.99	83.82	83.28
256	**Yes**	51.07	52.26	55.83	54.90	62.09	61.28	57.50	52.37	49.75

Table 5. *F1 Score of DOC-CrackNet for all the tested patch sizes and with or without augmentation*

Size	Augm	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
64	**No**	82.57	85.40	86.84	87.64	88.28	88.70	88.85	88.78	88.00
64	**Yes**	83.57	87.58	89.54	90.35	90.28	89.35	86.86	81.57	68.70
128	**No**	73.72	76.61	78.09	79.04	79.68	79.91	79.98	79.45	77.57
128	**Yes**	72.44	76.07	78.56	79.94	80.11	76.75	63.10	39.33	12.22
256	**No**	79.86	82.81	83.97	84.61	84.87	84.90	84.70	84.20	83.09
256	**Yes**	67.61	68.06	69.31	69.87	67.79	55.66	32.42	14.55	3.16

with distant background texture. Conversely, the larger 128×128 and 256×256 patches introduce excessive global noise and severe class imbalance (i.e., a vastly higher ratio of background concrete pixels to thin crack pixels). This negatively impacts the classification of the central pixel, particularly when spatial augmentations are applied. Therefore, the 64×64 patch size with data augmentation and a threshold of $\sigma = 0.5$ was selected as the optimal baseline for the integrated framework.

4.3. Orientation estimation

Once cracks were fully characterized, the procedure described in Section 3.6 was applied to the extracted patches for identifying their predominant direction. To this end, the effects of the erosion filter on three different masks are shown in Figure 6.

Subsequently, Figure 7 reports the results in terms of orientation, showing a blue line indicating the inclination and a legend reporting the angle and the desired output (i.e., vertical, diagonal, and horizontal).

4.4. Qualitative evaluation and discussion

As the last step in assessing the effectiveness of DOC-Crack, an experimental evaluation is provided by performing a qualitative evaluation of the results achieved by DOC-CrackNet-129 on a set of images that do not belong to the training set. The images shown in Figure 8 were selected to highlight some of the main pros and cons of DOC-Crack by showing different situations. Specifically, Figure 8(a) shows the results that can be achieved when a single and visible crack is present in the image. In this case, DOC-CrackNet-129 provided robust and consistent results, despite the varying angles of the crack depicted in the patch. The same conclusions could be drawn by looking at Figure 8(c), where the network was able to identify an irregular and discontinuous crack with high reliability. However, it is worth noting that the network can be misled under specific circumstances due to certain occurrences in the provided patch. For example, Figure 8(b) shows some irregularities due to the formwork of the structural element, which in the patch appeared visually similar to vertical cracks. Analogously, Figure 8(d) presents several irregularities in the

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

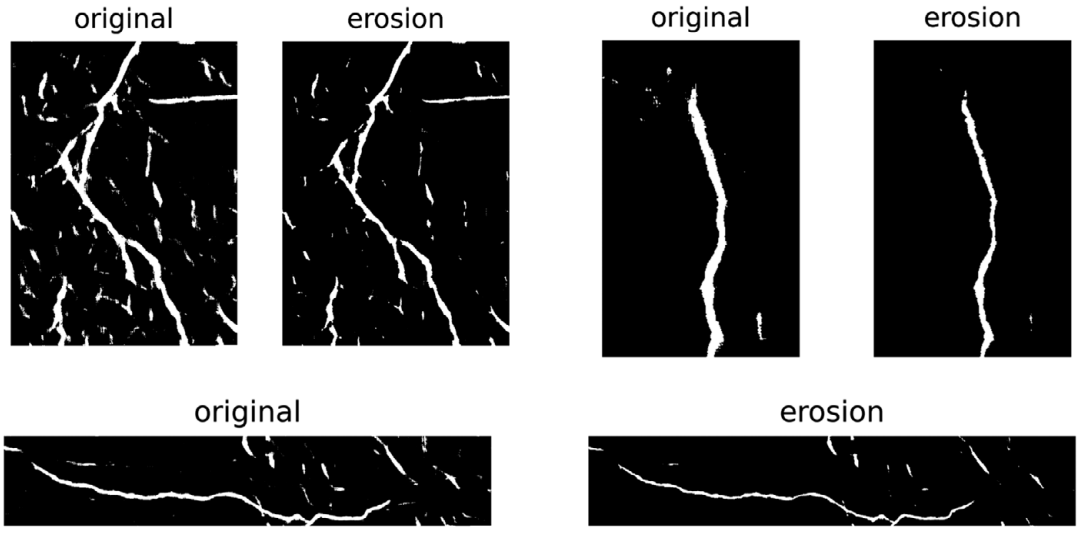


Figure 6. *Application of the erosion filter on three different masks. The general effect of applying the filter on the eroded masks caused an overall reduction of the noisy outliers.*

form of honeycombs and abrasions on the concrete surface, which could also be misinterpreted if the network is not provided with contextual information. These effects resulted in visual incoherence of the provided patches, as highlighted by the predictions provided by the network for both patches in Figure 8 (b) and (d). In such cases, while the network is still able to properly track the main crack in both situations, the prediction was affected by a certain degree of noise. Future evolution of the algorithm should address these situations by adopting specific solutions, such as data preprocessing to filter out noisy backgrounds, fusion with 3D surface data to identify defects such as honeycombs, or semantic enrichment via knowledge embedded into large language models to provide contextual information and, therefore, filter out formwork effects on the structural elements.

5. Application of DOC-Crack on a real case study

With the aim of testing the proposed procedure on a case study, DOC-Crack was used to investigate cracks on the deck of a real-life bridge. It is worth anticipating that, for privacy reasons, no registry information can be provided about the case study. The RC bridge is composed of frame piers and a deck characterized by three main beams. The focus of the investigation was oriented on the external beams, for which a survey through UAV was performed. In particular, a DJI Air 3 Fly More Combo, equipped with a 48MP camera, was used to take high-resolution images of the bridge. According to the procedure in Section 3.1, consecutive shots were performed (every 3 seconds) at a distance of around 1 meter, to completely survey different external beams, by ensuring a minimum overlapping zone 30%for the images. In Figure 9, the phase of the survey is reported.

Once the data were collected, the proposed procedure consisted of preparing images for the stitching phase. For the case at hand, three consecutive images were selected for two different bays (named in the following, Case 1 and Case 2). Obviously, more than three photos could be included in the analysis, but the authors' choice was aimed at reducing the computational effort in the image processing and to test DOC-Crack for more than one case (i.e., Case 1 and Case 2). Figures 10 and 11 show the three selected images and the related stitched image for Case 1 and Case 2, respectively, according to the approach reported in Section 3.2. It is worth observing that the stitching operation was performed by using as key points the holes in the beams (available for initial assembly of the bridge) and, as expected, the result was not affected by any distortion.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

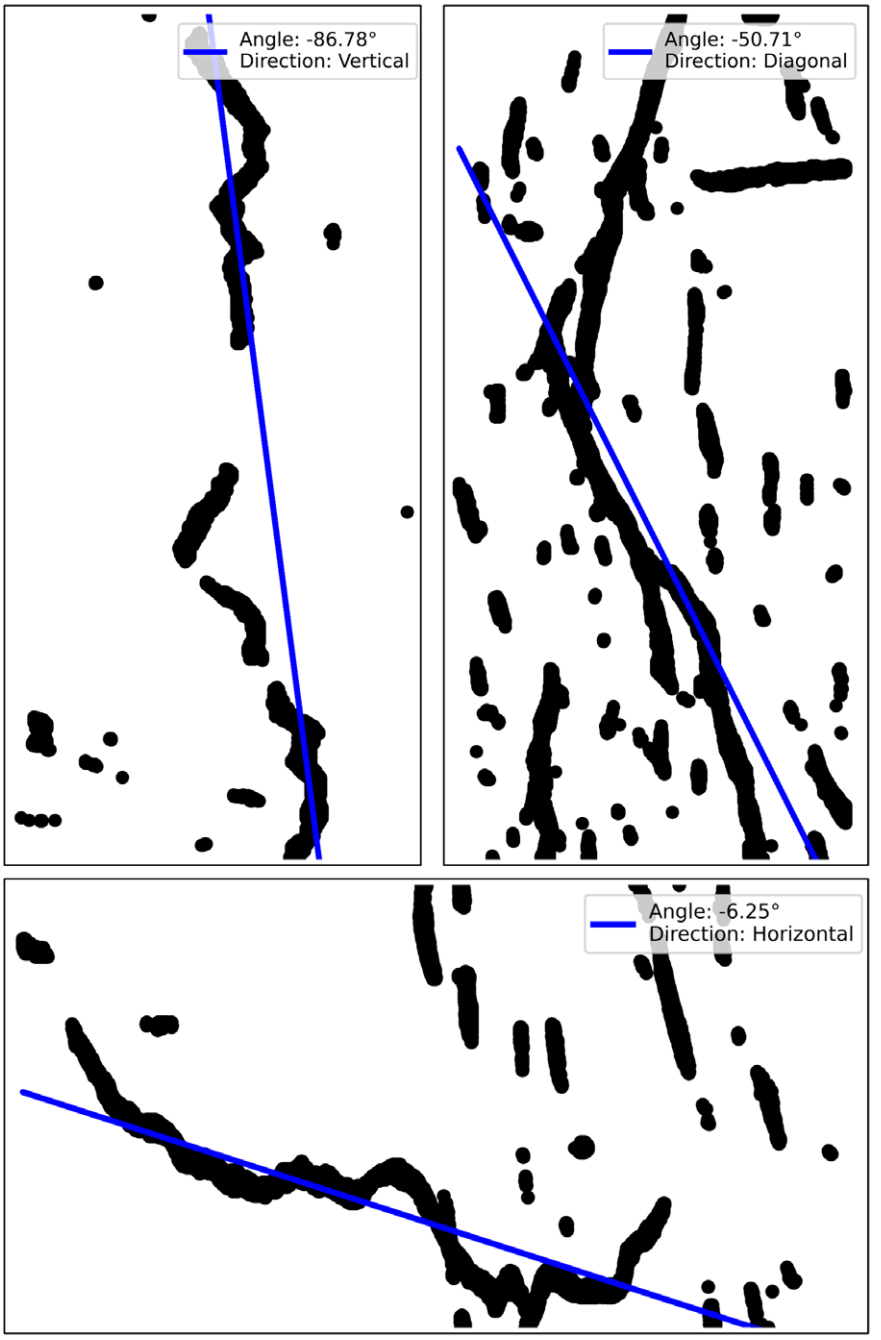


Figure 7. *The results of the orientation evaluation on the same masks are shown in Figure 6. As shown, the methodology could correctly evaluate the orientation of the crack in all three cases.*

The next step of the procedure consisted of crack detection on the stitched images. To this scope, as reported in Section 3.4, the trained version of YOLO11 was used, and the results can be observed for both Case 1 and Case 2 in Figure 12. From the analysis of the results, some aspects can be highlighted. First, as the primary objective of the proposed approach, the algorithm identifies almost all cracks, covering the

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

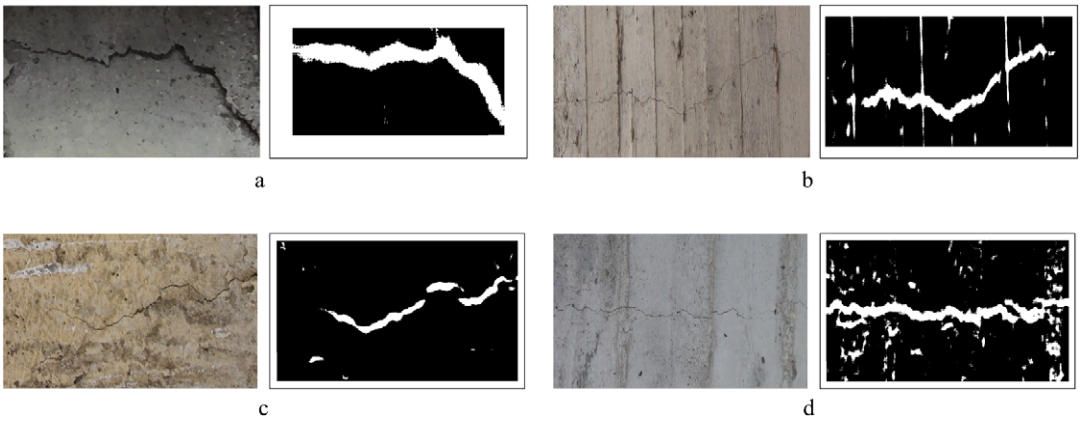


Figure 8. *Qualitative analysis. Each labeled subfigure represents a different crack in the dataset (on the left), with the associated crack identified by CrackNet (on the right).*



Figure 9. *Photos of the survey phase, performed through UAV flight.*

effective crack pattern in the stitched image. After each bounding box provides a confidence score (a number close to the bounding box), which indicates the reliability of the prediction. In fact, for very thin cracks, albeit with a low confidence score, the algorithm is able to identify the considered defect, often also difficult to recognize by the human eye. Finally, as the main advantage of the trained algorithm, the proposed detector is able to distinguish the cracks from other parts of the images, such as the release agent used at the time of construction (i.e., the near-vertical darker lines in the figures), to confirm the goodness of the prediction (differently from the formwork lines in Figure 8). As a main disadvantage, the detector tends to subdivide long cracks into shorter ones, in accordance with the performed training. If from one hand, this can represent an issue for the proposed approach (especially for the counting), on the other hand, the cases in which some cracks are missed by the detector can be easily recognized by the user.

After performing crack detection in the stitched image, the algorithm of crack segmentation, DOC-CrackNet, was applied. In particular, the model with higher performance, DOC-CrackNet-129 (defined in Section 4.1), was used for all detected cracks, and some results can be observed in Figure 13. In this latter, referring to six cases randomly chosen, the first row reports the cracks previously detected in the stitched image, while the second row reports the obtained estimation masks, assuming σ equal to 0.5. It is worth noting that the results in terms of estimation masks are reported as already elaborated with the erosion filter, in order to estimate the orientation (see later). Still, the third row highlights with a green patch the detected cracks, for the purpose of crack pattern definition to be reported on the original stitched image.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.



Figure 10. *Case 1: A set of three consecutive shots and a stitched image.*



Figure 11. *Case 2: A set of three consecutive shots and a stitched image.*

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.



Figure 12. *Crack detection on Case 1 and Case 2, respectively.*

The obtained results show a certain flexibility of DOC-CrackNet-129, which is able to detect different types of cracks, characterized by different thicknesses. In addition, a few noise effects can be observed to confirm the limitation mentioned in Section 4.4.

Finally, the last step of the proposed procedure consists of defining the cracks' orientation. According to Section 4.3, the proposed algorithm was used, and the obtained results are reported in Figure 14. Using the ranges defined in Table 2, the evaluation of the inclination degree is reported. Figure 14 shows, on the same set of cracks reported in Figure 13, that three cracks were classified as vertical and three were classified as diagonal.

In the end, the outcome of DOC-Crack is the crack pattern, which is shown in Figure 15. The obtained results suggest the importance of tools like DOC-Crack, which provides valuable information to users (e.g., surveyors of road management companies), and suggests possible interventions in view of risk mitigation.

6. Discussion

6.1. Manual *versus* automatic crack identification

As the final step of the work, it is interesting to assess the results of DOC-Crack against manual crack annotation. To this end, one of the two cases studied was selected, namely the stitched image of Case 1 (Figure 10). In this case, the authors—acting as domain experts in structural engineering—manually

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

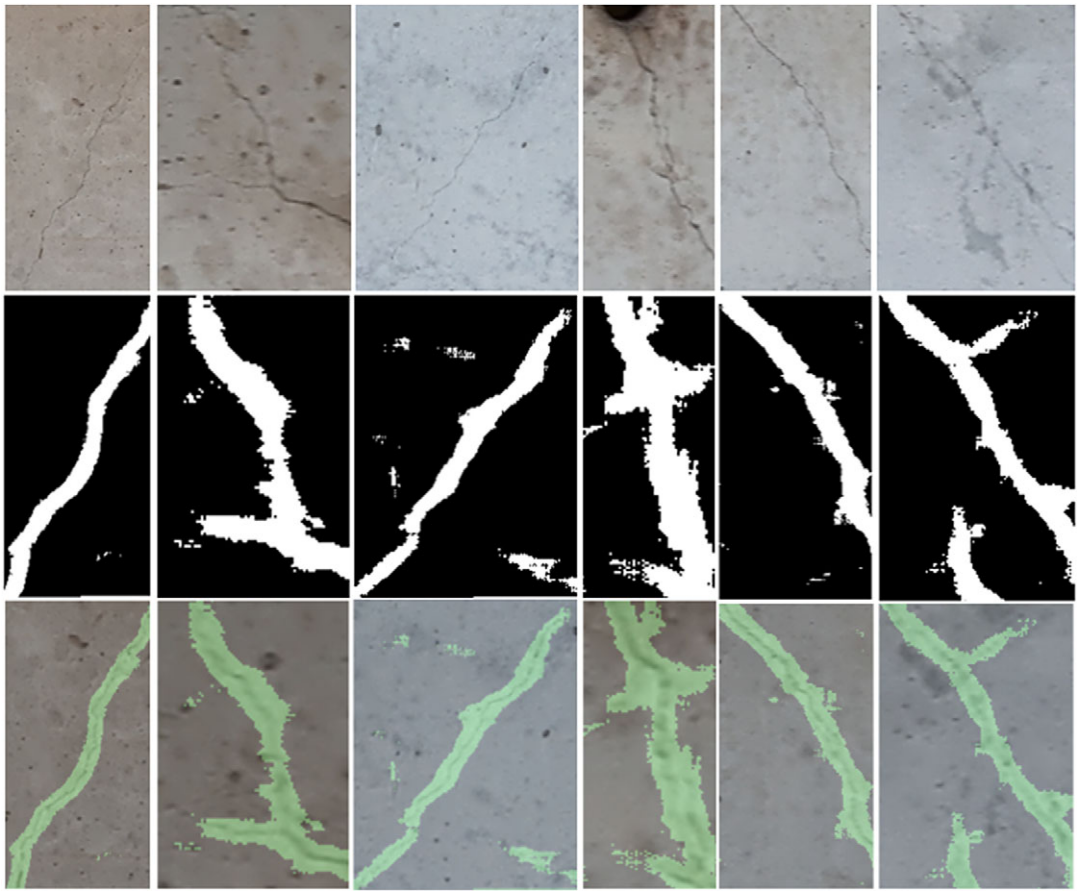


Figure 13. *Results of DOC-CrackNet.*

identified visible cracks using red traces, as shown in Figure 16. Comparing the capacity of the human eye against the output of DOC-Crack (see Figure 15) highlights several key aspects.

Regarding crack detection, as previously noted, DOC-Crack tends to subdivide long, continuous cracks into shorter segments whenever visual discontinuities (e.g., surface noise or patch borders) occur. Conversely, the human eye inherently extrapolates the entire crack development across the surface of the element. This discrepancy leads to different final counts: the automated detection identified 11 short cracks, while manual detection identified nine continuous cracks (plus one branching off in the central part). While simple heuristic postprocessing rules, such as merging adjacent detections based on distance proximity and orientation continuity, can partially mitigate this over-counting, they often fail or introduce false connections in dense, highly branched *spider-web* crack patterns. Therefore, this counting artifact is acknowledged as a current limitation of the pixel-level segmentation approach. Future developments will investigate transitioning to true instance-level segmentation to organically map complex, multibranching cracks as unified instances without manual parameter tuning.

As for crack orientation, according to the geometric rules established for the automated computation, both the manual and automatic approaches returned consistent results for the main crack branches.

6.2. Limitations

It is important to underline that the proposal uses standard feature-based homography to generate the composite panoramas of the RC bridge. However, the overall reliability of the proposed framework is

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

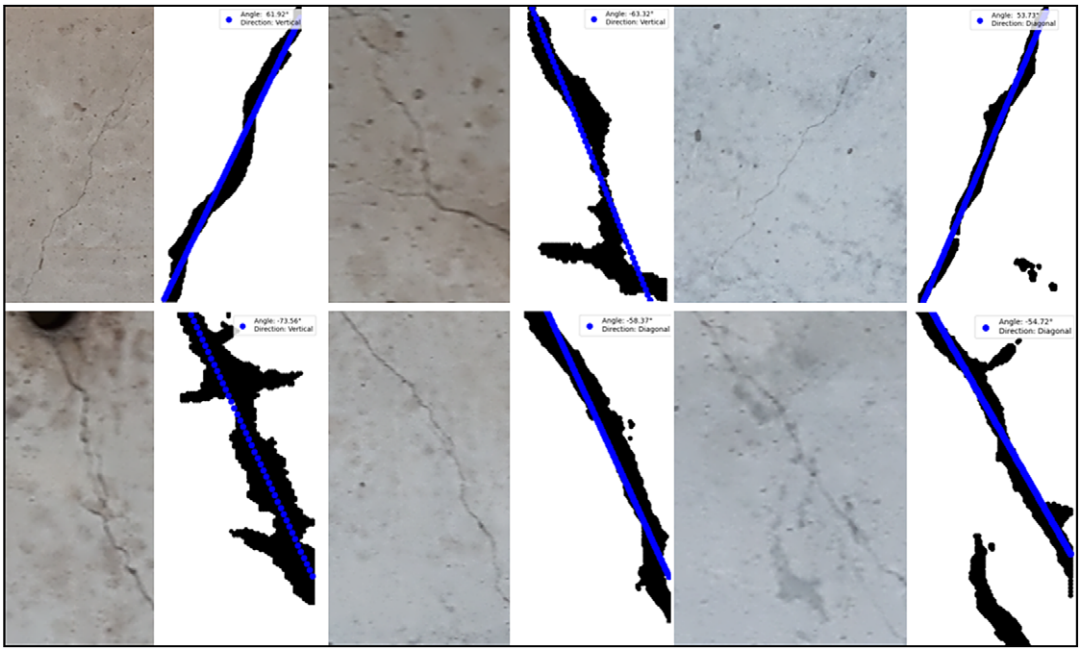


Figure 14. *Results in terms of orientation.*

based on specific operational assumptions during image acquisition. Specifically, the camera should maintain a roughly perpendicular optical axis to the surface, and there should be a high degree of overlap between consecutive frames. Furthermore, as the homography mathematically assumes that the target surface is a bidimensional planar element, it may become insufficient when inspecting nonplanar elements, such as cylindrical piles, introducing local distortions or misalignments during stitching. These artifacts are particularly detrimental, as they can visually fracture continuous crack patterns into disjunct segments, thereby artificially inflating the automated crack count and distorting the orientation mapping. To overcome these limitations, future work could replace the bidimensional stitching module with 3D photogrammetry to be applied before the detection pipeline.

Furthermore, it is important to underline how assigning a single, dominant orientation via linear fitting is an inherent limitation when dealing with irregularly branched cracks. Hence, while the proposal fitting proved reliable for the relatively straight crack patterns in the current case study, representing a complex, multidirectional crack as a single linear vector oversimplifies its local geometry. Future iterations of the framework will explore instance-level skeletonization to map varying orientations along the continuous path of a single curved crack.

Finally, it is important to note that the framework may exhibit partially overlapping cracks or cracks that are partially split into multiple segments. To this end, several approaches can be integrated into future works. For example, the nonmaximum suppression step of a YOLO-based algorithm can be enhanced to consider proximity constraints, therefore merging multiple nonoverlapping bounding boxes. This would allow us to maintain proper performance while avoiding instance segmentation techniques, which may entail a higher computational cost.

7. Conclusions

The paper presents an automated framework for characterizing crack patterns in RC bridges. The proposed tool, named DOC-Crack, is capable of detecting, counting, and orienting cracks in structural elements. The pipeline consists of various stages. The first step involves data collection, where a stack of

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

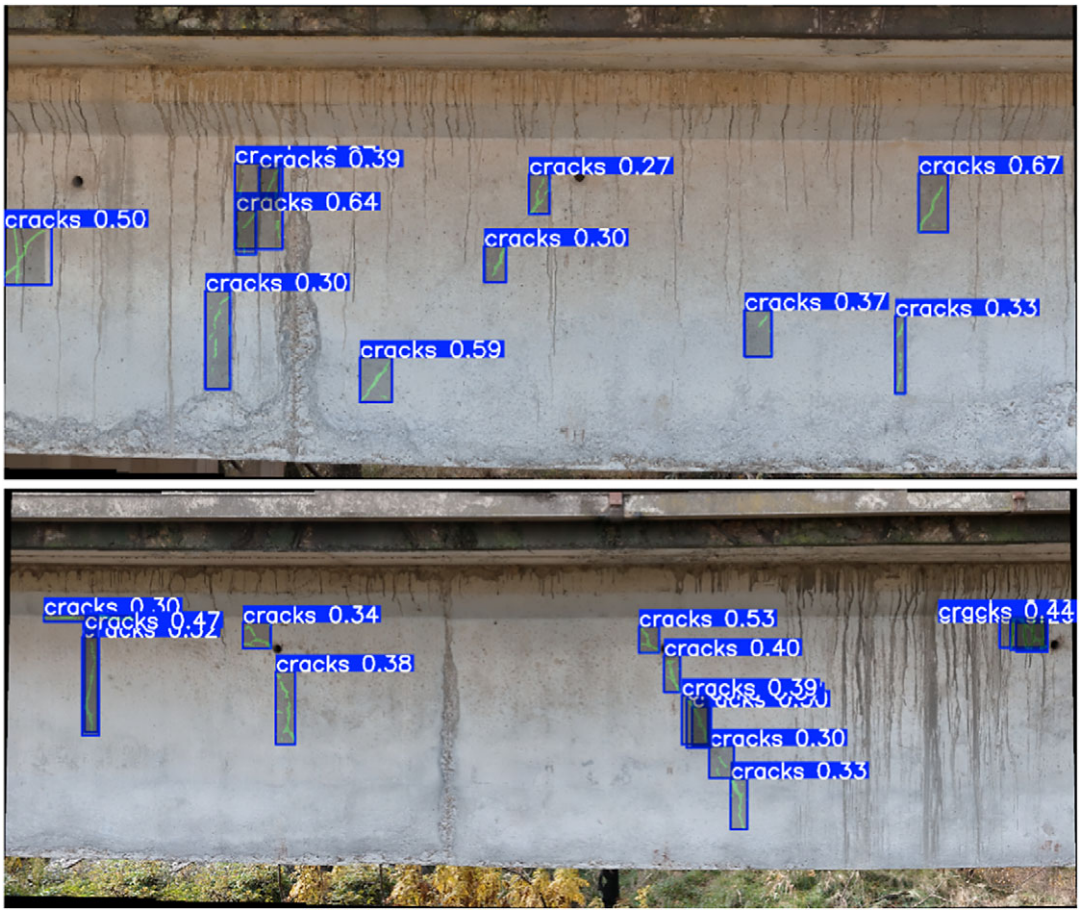


Figure 15. *Results of DOC-Crack on Case 1 and Case 2, respectively, showing crack patterns.*



Figure 16. *Case 1: Results of manual annotation.*

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

photos for a structural element to inspect is collected, ensuring an overlap between consecutive images. In the second step, a phase of image stitching is proposed, which enables the acquisition of an image encompassing the near-full structural element to be investigated. The third step aims to elaborate a pipeline of DL algorithms, including an improved object detector specifically trained on a dataset of cracks, and a pixel-based segmentation algorithm to identify cracks on the output of the detection. Based on the elaborations performed, DOC-Crack enables the counting of cracks and the definition of their related orientation. The output of DOC-Crack is the crack pattern of the inspected structural element, which can serve as support for domain experts in assessing the current state of the structure and predicting possible future issues, as well as planning related interventions.

As shown throughout the manuscript, various advantages can be mentioned regarding DOC-Crack. Specifically, the proposed image stitching enables the reconstruction of the overall appearance of the RC element by exploiting invariant local features, thereby providing a comprehensive perspective of the element under investigation. Afterwards, the segmentation algorithm led to a per-crack evaluation, providing the effective trace and orientation of each visible crack. Overall, DOC-Crack offers several advantages, including its modularity, which allows each step to be independently deployed and used as needed. In addition, the proposed approach offers an aseptic visual information of cracks on the element to domain experts. The real aim of the authors is to provide an output that must be examined by the structural engineer, who can perform the necessary evaluations and make decisions. As a main disadvantage, while the network is able to identify and track the main cracks in the figures, some predictions can be affected by noise, which can result in an inaccurate final crack pattern. Still, although almost all cracks are detected, long cracks could be subdivided into smaller ones, biasing the final count but not affecting the crack pattern. Future evolutions of the algorithm should address these situations by adopting specific solutions, such as data preprocessing to filter out noisy backgrounds, fusion with 3D surface data to identify defects such as honeycombs, semantic enrichment via knowledge embedded into large language models to provide contextual information, and, therefore, filter out formwork effects on the structural elements. In addition, future developments should leverage the intrinsic parallelizability of the proposed pipeline, which, with an optimized implementation, could provide real-time capabilities on constrained hardware, such as low-end system-on-a-chip or smartphones.

Data availability statement. The data supporting the findings are available at https://doi.org/10.5281/zenodo.18218986 Cardellicchio et al. (2026). The other datasets can be found in the referenced publications.

Acknowledgements. All authors thank Prof. Angelo Doglioni for the UAV survey.

Author contribution. Conceptualization: A.C., V.R., S.R.; Data curation: A.C., V.R., V.M.D.M.; Data visualization: A.C., V.R.; Funding: G.U.; Methodology: A.C., V.R., V.M.D.M.; Review original draft: Ag.N., V.R., An.N., S.R., G.U.; Writing original draft: A.C, S.R. All authors approved the final submitted draft.

Funding statement. S.R. thanks funding by Fabre Consortium, within the research grant "Sviluppo e implementazione di strategie basate sulla Intelligenza Artificiale per l'analisi e il monitoraggio del rischio strutturale di ponti e viadotti esistenti." An.N. and G.U. thank funding by Centro Nazionale Sustainable Mobility Center, within the framework of "MOST" project, CUP code: D93C22000410001.

Competing interests. The authors declare none.

Ethical standard. The research adheres to all ethical guidelines, including compliance with the legal requirements of the study country.

References

Anisha A, **Jacob A**, **Davis R and Mangalathu S** (2022) Fragility functions for highway rc bridge under various flood scenarios. *Engineering Structures 260*, 114244. https://doi.org/10.1016/j.engstruct.2022.114244.

Bazzucchi F, **Restuccia L and Ferro GA** (2018) Considerations over the italian road bridge infrastructure safety after the Polcevera Viaduct collapse: Past errors and future perspectives. *Frattura e Integrita Strutturale 12*(46), 400–421. https://doi.org/10.3221/IGF-ESIS.46.37.

Calò M, **Ruggieri S**, **Nettis A and Uva G** (2024) A mtinsar-based early warning system to appraise deformations in simply supported concrete girder bridges. *Structural Control and Health Monitoring 2024*, 8978782.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Calvi GM, **Moratti M**, **O'Reilly GJ**, **Scattarreggia N**, **Monteiro R**, **Malomo D**, **Calvi PM and Pinho R** (2019) Once upon a time in Italy: The tale of the morandi bridge. *Structural Engineering International 29*(2), 198–217. https://doi.org/10.1080/10168664.2018.1558033.

Cardellicchio A, **Ruggieri S**, **Leggieri V and Uva G** (2022) View vulma: Data set for training a machine-learning tool for a fast vulnerability analysis of existing buildings. *Data 7*(1), 4.

Cardellicchio A, **Ruggieri S**, **Nettis A**, **Renò V and Uva G** (2023) Physical interpretation of machine learning-based recognition of defects for the risk management of existing bridge heritage. *Engineering Failure Analysis 149*, 107237. 10.1016/j.engfailanal.2023.107237.

Cardellicchio A, **Ruggieri S**, **Nettis A**, **Renò V**, **Di Mucci VM and Uva G** (2026) Doc-Crack Dataset. Version 1.0. https://doi.org/10.5281/zenodo.18218986.

Cha Y-J, **Choi W and Büyüköztürk O** (2017) Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering 32*(5), 361–378. https://doi.org/10.1111/mice.12263.

Chu H, **Wang W and Deng L** (2022) Tiny-crack-net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks. *Computer-Aided Civil and Infrastructure Engineering 37*(14), 1914–1931. https://doi.org/10.1111/mice.12881.

Datumaro (2022). Dataset management framework (datumaro). Available at https://github.com/open-edge-platform/datumaro.

Deng J, **Lu Y and Lee VC-S** (2020) Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering 35*(4), 373–388. https://doi.org/10.1111/mice.12497.

Deng L, **Sun T**, **Yang L and Cao R** (2023) Binocular video-based 3d reconstruction and length quantification of cracks in concrete structures. *Automation in Construction 148*, 105719.

Di Mucci VM, **Cardellicchio A**, **Ruggieri S**, **Nettis A**, **Renò V and Uva G** (2024) Artificial intelligence in structural health management of existing bridges. *Automation in Construction 167*, 105719. https://doi.org/10.1016/j.autcon.2024.105719.

Ding W, **Yang H**, **Yu K and Shu J** (2023) Crack detection and quantification for concrete structures using UAV and transformer. *Automation in Construction 152*, 104929. https://doi.org/10.1016/j.autcon.2023.104929.

Fischler MA and Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*(6), 381–395. https://doi.org/10.1145/358669.358692.

Guo J, **Liu P**, **Xiao B**, **Deng L and Wang Q** (2024) Advanced bridge visual inspection using real-time machine learning in edge devices. *Automation in Construction 158*, 105186.

Harris C and Stephens M (1988) A combined corner and edge detector. In *Alvey Vision Conference*. Manchester: Citeseer, pp. 10–5244.

He K, **Zhang X**, **Ren S and Sun J** (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV: IEEE, pp. 770–778.

Jiang S and Zhang J (2019) Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. *Computer-Aided Civil and Infrastructure Engineering 35*, 549–564.

Jocher G and Qiu J (2024) Ultralytics Yolo11. Available at https://www.ultralytics.com/.

Kang DH and Cha Y-J (2021) Efficient attention-based deep encoder and decoder for automatic crack segmentation. *Structural Health Monitoring 21*, 2190–2205.

Kim H, **Sim S-H and Spencer B** (2022) Automated concrete crack evaluation using stereo vision with two different focal lengths. *Automation in Construction 135*, 104136. https://doi.org/10.1016/j.autcon.2022.104136.

Laflamme S, **Blasch E**, **Ubertini F**, **Liu Z**, **Wertz J**, **Knott C**, **Cherry M**, **Lindgren E**, **Chang F-K**, **Kumar A**, **Poole J**, **Worden K**, **Downey A**, **Wei J**, **Musgrave PF**, **Wong AS**, **Quaranta G**, **Rosso MM**, **Marano GC**, **Chen Y**, **Ardiles-Cruz E**, **Soleimani-Babakamali MH**, **Avci O**, **Inman DJ**, **Taciroglu E**, **Dodson J**, **Chen G**, **Meng W**, **Zhu C**, **Liu Z**, **Zuo J**, **Liu Q**, **Khan S**, **Hu C**, **Hu Z**, **Cicirello A**, **Cross E**, **Chatzi E**, **Weng Y**, **Yuan J**, **Wen S**, **Han L**, **Metaxas D**, **Tronci EM**, **Moaveni B**, **Chen Q**, **Ng MS**, **Hackl J**, **Chen G**, **Wei S**, **Mitoulis S-A**, **Izonin I**, **Uva G**, **Ruggieri S**, **Mao Z**, **Kiranyaz S**, **Devecioglu OC**, **Gabbouj M and Mohammadi J** (2026) Roadmap: Integrating artificial intelligence in structural health monitoring systems. *Measurement Science and Technology 37*(10), 103001. https://doi.org/10.1088/1361-6501/ae3abb.

Li S, **Zhao X and Zhou G** (2019) Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering 34*(7), 616–634. https://doi.org/10.1111/mice.12433.

Liang X (2019) Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with bayesian optimization. *Computer-Aided Civil and Infrastructure Engineering 34*(5), 415–430. https://doi.org/10.1111/mice.12425.

Liu Y-F, **Nie X**, **Fan J-S and Liu X-G** (2019) Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction. *Computer-Aided Civil and Infrastructure Engineering 35*(5), 511–529.

Liu Y, **Shao Z and Hoffmann N** (2021) Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. arXiv preprint arXiv:2112.05561.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

MIT (2021) Linee guida per la gestione del rischio dei ponti esistenti e delle Istruzioni operative per l'applicazione delle Linee Guida stesse. Available at https://www.mit.gov.it/nfsmitgov/files/media/normativa/2025-11/Istr.Operative_LG_Ponti_ANSFISA_rev_04_08_2025.pdf.

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.

Nettis A, **Iacovazzo P**, **Raffaele D**, **Uva G and Adam JM** (2022) Displacement-based seismic performance assessment of multi-span steel truss bridges. *Engineering Structures 254*, 113832. https://doi.org/10.1016/j.engstruct.2021.113832.

Nettis A, **Massimi V**, **Nutricato R**, **Nitti DO**, **Samarelli S and Uva G** (2023) Satellite-based interferometry for monitoring structural deformations of bridge portfolios. *Automation in Construction 147*, 104707. https://doi.org/10.1016/j.autcon.2022.104707.

Nettis A, **Nettis A**, **Ruggieri S and Uva G** (2024) Corrosion-induced fragility of existing prestressed concrete girder bridges under traffic loads. *Engineering Structures 314*, 118302. https://doi.org/10.1016/j.engstruct.2024.118302.

Park S, **Eem S-H and Jeon H** (2020) Concrete crack detection and quantification using deep learning and structured light. *Construction and Building Materials 252*, 119096. https://doi.org/10.1016/j.conbuildmat.2020.119096.

Paszke A, **Gross S**, **Massa F**, **Lerer A**, **Bradbury J**, **Chanan G**, **Killeen T**, **Lin Z**, **Gimelshein N**, **Antiga L**, **Desmaison A**, **Köpf A**, **Yang E**, **DeVito Z**, **Raison M**, **Tejani A**, **Chilamkurthy S**, **Steiner B**, **Fang L**, **Bai J and Chintala S** (2019) Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems 32*, 8026–8037.

Provost F (2000) Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 Workshop on Imbalanced Data Sets*, Vol. *68*. Austin, TX: AAAI Press, pp. 1–3.

Redmon J, **Divvala S**, **Girshick R and Farhadi A** (2016) You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, NV: IEEE, pp. 779–788.

Ruggieri S, **Cardellicchio A**, **Nettis A**, **Renò V and Uva G** (2023) Using machine learning approaches to perform defect detection of existing bridges. *Procedia Structural Integrity 44*, 2028–2035. XIX ANIDIS Conference, Seismic Engineering in Italy. https://doi.org/10.1016/j.prostr.2023.01.259.

Ruggieri S, **Cardellicchio A**, **Nettis A**, **Renò V and Uva G** (2024) Automatic detection of typical defects in reinforced concrete bridges via yolov5. *Procedia Structural Integrity 62*, 129–136. II Fabre Conference – Existing bridges, viaducts and tunnels: research, innovation and applications (FABRE24). https://doi.org/10.1016/j.prostr.2024.09.025.

Ruggieri S, **Cardellicchio A**, **Nettis A**, **Renò V and Uva G** (2025) Using attention for improving defect detection in existing RC bridges. *IEEE Access 13*, 18994–19015. https://doi.org/10.1109/ACCESS.2025.3532832.

Saleem MR, **Park J-W**, **Lee J**, **Jung H-J and Sarwar M** (2020) Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network. *Structural Health Monitoring 20*(4), 1760–1777.

Sekachev B, **Manovich N**, **Zhiltsov M**, **Zhavoronkov A**, **Kalinin D**, **Hoff B**, **Osmanov T**, **Kruchinin D**, **Zankevich A**, **Markelov M**, **Chenuet M**, **Melnikov A**, **Kim J**, **Ilouz L**, **Glazov N**, **Tehrani R**, **Jeong S**, **Skubriev V**, **Yonekura S and Truong T** (2020) opencv/cvat: v1.1.0. Available at https://opencv.org/.

Spencer BF, **Hoskere V and Narazaki Y** (2019) Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering 5*(2), 199–222. https://doi.org/10.1016/j.eng.2018.11.030.

Wang W and Su C (2022) Automatic concrete crack segmentation model based on transformer. *Automation in Construction 139*, 104275. https://doi.org/10.1016/j.autcon.2022.104275.

Wang H-F, **Zhai L**, **Huang H**, **Guan L-M**, **Mu K-N and Wang G-p** (2020) Measurement for cracks at the bottom of bridges based on tethered creeping unmanned aerial vehicle. *Automation in Construction 119*, 103330. https://doi.org/10.1016/j.autcon.2020.103330.

Wang X, **Demartino C**, **Narazaki Y**, **Monti G and Spencer BF** (2023) Rapid seismic risk assessment of bridges using UAV aerial photogrammetry. *Engineering Structures 279*, 115589. https://doi.org/10.1016/j.engstruct.2023.115589.

Yang L, **Zhang R**, **Li L and Xie X** (2021) SimAM: A simple, parameter-free attention module for convolutional neural networks.. In *Proceedings of the 38th International Conference on Machine Learning*, vol. *139*, pp. 11863–11874. Available at https://proceedings.mlr.press/v139/yang21o.html.

Yu Z, **Shen Y and Shen C** (2021) A real-time detection approach for bridge cracks based on yolov4-fpm. *Automation in Construction 122*, 103514. https://doi.org/10.1016/j.autcon.2020.103514.

Cite this article: Cardellicchio A, Renò V, Natali A, Di Mucci VM, Nettis A, Ruggieri S and Uva G (2026). An automated framework to characterize crack patterns in existing RC bridges. *Data-Centric Engineering*, 7, e26. doi:10.1017/dce.2026.10060

Downloaded from https://www.cambridge.org/core. 17 Nov 2026 at 16:03:13, subject to the Cambridge Core terms of use.